

Woebot_notebookRun

July 5, 2020

```
[1]: # Importing Main Libraries
import torch

torch.cuda.current_device()
import numpy as np
from torch.utils.data import DataLoader
from preprocess_load_embedding import EmpathyDataset, DataPreprocessing
from sklearn.model_selection import train_test_split
import torch.backends.cudnn as cudnn
from models_LSTMs import LSTM_fixed_len, LSTM_variable_input, LSTM_glove_vecs
from train_test_lossCriterion import train, get_optimizer_criterion_scheduler
import warnings

warnings.filterwarnings('ignore')
np.random.seed(1)

# Model was trained using GPU in CUDA Environment
print("Cuda Available: {}".format(torch.cuda.is_available()))

# File Names
labelled_message_file = "/media/HDD_2TB.1/machine-learning-engineer/
↳labeled_messages.csv"
empathies_file = "/media/HDD_2TB.1/machine-learning-engineer/empathies.csv"
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]      /home/akashdevgun/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Cuda Available: True

```
[2]: # Method that calls to train different types of LSTMs
def train_lstm(model, num_epochs, learning_rate, loss_weights, device,
↳train_queue, valid_queue):
    model = model.to(device)
    criterion, optimizer, scheduler = get_optimizer_criterion_scheduler(model,
↳num_epochs, learning_rate,
```

```

→loss_weights, device)

    for epoch in range(num_epochs):
        scheduler.step()
        train(model, device, train_queue, valid_queue, optimizer, epoch,
→criterion)

```

```

[3]: # Main Method
def main():
    # Object 'Data' is created by Class Named -> 'DataPreprocessing' with file
→names as parameters
    data = DataPreprocessing(labelled_message_file, empathies_file)

    # Method Call describes number of words in Corpus, messages lengths
    data.describe_counts()

    # Method call for MultiLabel Encoding, for weighted label weights to handle
→imbalance
    output_size, loss_weights = data.label_binarizer_get_weights()

    # Call to Get X and Y
    X = data.get_X_data()
    y = data.get_Y_data()
    print('\n')

    # Train and Test Split
    X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.3)

    # Baseline Classifier using SVC to calculate Accuracy and Area Under Curve
→Scores
    print("***** Baseline AUC Scores *****")
    acc_svm, roc_svm = data.modelling("SVC", X_train, X_valid, y_train, y_valid)
    print("SVM Modelling --> Validation Acc. : %.3f, Validation AUC Score : %.
→3f" % (acc_svm, roc_svm))
    acc_RF, roc_RF = data.modelling("RandomForest", X_train, X_valid, y_train,
→y_valid)

    print("***** Statistical Method better then Baseline *****")
    # Baseline Classifier using SVC to calculate Accuracy and Area Under Curve
→Scores
    print("Random Forest Modelling --> Validation Acc. : %.3f, Validation AUC
→Score : %.3f" % (acc_RF, roc_RF))

    # Class 'EmpathyDataset' Called for train and valid dataset to load while
→run time during training and testing
    train_ds = EmpathyDataset(X_train, y_train)

```

```

valid_ds = EmpathyDataset(X_valid, y_valid)

vocab_size = len(data.words)
num_epochs = 1001
batch_size = 1000
learning_rate = 0.3

# Data Loader for train and test
train_queue = DataLoader(train_ds, batch_size=batch_size, shuffle=True)
valid_queue = DataLoader(valid_ds, batch_size=batch_size, shuffle=False)

# CUDA Environment Settings
torch.cuda.set_device(0)
cudnn.benchmark = True
torch.manual_seed(1)
cudnn.enabled = True
torch.cuda.manual_seed(1)
use_cuda = torch.cuda.is_available()
device = torch.device("cuda" if use_cuda else "cpu")

# LSTMs models with fixed length Input, variable length Input, using
→StandFord Glove Representations
print('\n')
print('-----LSTMs Fixed Length Input-----')
model1 = LSTM_fixed_len(vocab_size, 48, 96, output_size)
train_lstms(model1, num_epochs, learning_rate, loss_weights, device,
→train_queue, valid_queue)

print('\n')
print('-----LSTMs Variable Length Input-----')
model2 = LSTM_variable_input(vocab_size, 48, 96, output_size)
train_lstms(model2, num_epochs, learning_rate, loss_weights, device,
→train_queue, valid_queue)

print('\n')
print('-----LSTMs with Glove Representation of Input-----')
word_vecs = data.load_glove_vectors()
pretrained_weights, vocab, vocab2index = data.get_emb_matrix(word_vecs)
model3 = LSTM_glove_vecs(vocab_size, 50, 96, pretrained_weights, output_size)
train_lstms(model3, num_epochs, learning_rate, loss_weights, device,
→train_queue, valid_queue)

```

```

[4]: if __name__ == '__main__':
      main()

```

Data Shape: (3562, 4)

First 5 Columns of Data:

num_seen	message	empathy ignore
----------	---------	----------------

0	2884	tired	tired	NaN
1	253	exhausted	tired	NaN
2	61	drained	tired	NaN
3	31	tired but happy	tired, happy	NaN
4	30	i am tired	tired	NaN

Number of words in Corpus: 2114

Message Avg Length : 5.123526108927569, Message Max Length : 121

Number of Empathies: 62, Output Shape is: (3562, 62)

First 5 Columns of Data After Preprocessing and MultiLabel Encoding:

	num_seen	message	empathy	message_length \
0	2884	tired	tired	1
1	253	exhausted	tired	1
2	61	drained	tired	1
3	31	tired but happy	tired, happy	3
4	30	i am tired	tired	3

	encoded	y_encoded \
0	[[2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[tired]
1	[[3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[tired]
2	[[4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[tired]
3	[[2, 5, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[tired, happy]
4	[[7, 8, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[tired]

	y_encoded_int
0	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
1	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
2	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
3	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
4	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...

***** Baseline AUC Scores *****

SVM Modelling --> Validation Acc. : 0.972, Validation AUC Score : 0.606

***** Statistical Method better then Baseline *****

Random Forest Modelling --> Validation Acc. : 0.892, Validation AUC Score : 0.685

-----LSTMs Fixed Length Input-----

Epoch: 1, Train loss: 0.676, Val loss: 0.668, Val Acc: 0.621, Val AUC_ROC Macro: 0.488, Val AUC_ROC Weighted : 0.488, Val Recall Macro: 0.474, Val Precision Weighted : 0.955, Val Ham Loss : 0.379

Epoch: 101, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.634, Val AUC_ROC Weighted : 0.634, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

Epoch: 201, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.677, Val AUC_ROC Weighted : 0.677, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

Epoch: 301, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.695, Val AUC_ROC Weighted : 0.695, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

Epoch: 401, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.704, Val AUC_ROC Weighted : 0.704, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

Epoch: 501, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.706, Val AUC_ROC Weighted : 0.706, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

Epoch: 601, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.709, Val AUC_ROC Weighted : 0.709, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

Epoch: 701, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.711, Val AUC_ROC Weighted : 0.711, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

Epoch: 801, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.712, Val AUC_ROC Weighted : 0.712, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

Epoch: 901, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.712, Val AUC_ROC Weighted : 0.712, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

Epoch: 1001, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.713, Val AUC_ROC Weighted : 0.713, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

-----LSTMs Variable Length Input-----

Epoch: 1, Train loss: 0.683, Val loss: 0.676, Val Acc: 0.530, Val AUC_ROC Macro: 0.501, Val AUC_ROC Weighted : 0.501, Val Recall Macro: 0.498, Val Precision Weighted : 0.957, Val Ham Loss : 0.470

Epoch: 101, Train loss: 0.011, Val loss: 0.011, Val Acc: 0.978, Val AUC_ROC Macro: 0.488, Val AUC_ROC Weighted : 0.488, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

Epoch: 201, Train loss: 0.009, Val loss: 0.010, Val Acc: 0.978, Val AUC_ROC Macro: 0.487, Val AUC_ROC Weighted : 0.487, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

Epoch: 301, Train loss: 0.009, Val loss: 0.009, Val Acc: 0.978, Val AUC_ROC Macro: 0.487, Val AUC_ROC Weighted : 0.487, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

Epoch: 401, Train loss: 0.008, Val loss: 0.009, Val Acc: 0.978, Val AUC_ROC Macro: 0.488, Val AUC_ROC Weighted : 0.488, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

Epoch: 501, Train loss: 0.008, Val loss: 0.009, Val Acc: 0.978, Val AUC_ROC Macro: 0.490, Val AUC_ROC Weighted : 0.490, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022
Epoch: 601, Train loss: 0.008, Val loss: 0.009, Val Acc: 0.978, Val AUC_ROC Macro: 0.491, Val AUC_ROC Weighted : 0.491, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022
Epoch: 701, Train loss: 0.008, Val loss: 0.009, Val Acc: 0.978, Val AUC_ROC Macro: 0.492, Val AUC_ROC Weighted : 0.492, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022
Epoch: 801, Train loss: 0.008, Val loss: 0.009, Val Acc: 0.978, Val AUC_ROC Macro: 0.492, Val AUC_ROC Weighted : 0.492, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022
Epoch: 901, Train loss: 0.008, Val loss: 0.009, Val Acc: 0.978, Val AUC_ROC Macro: 0.493, Val AUC_ROC Weighted : 0.493, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022
Epoch: 1001, Train loss: 0.008, Val loss: 0.009, Val Acc: 0.978, Val AUC_ROC Macro: 0.493, Val AUC_ROC Weighted : 0.493, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

-----LSTMs with Glove Representation of Input-----

Epoch: 1, Train loss: 0.682, Val loss: 0.674, Val Acc: 0.579, Val AUC_ROC Macro: 0.523, Val AUC_ROC Weighted : 0.523, Val Recall Macro: 0.528, Val Precision Weighted : 0.959, Val Ham Loss : 0.421
Epoch: 101, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.614, Val AUC_ROC Weighted : 0.614, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022
Epoch: 201, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.657, Val AUC_ROC Weighted : 0.657, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022
Epoch: 301, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.679, Val AUC_ROC Weighted : 0.679, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022
Epoch: 401, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.698, Val AUC_ROC Weighted : 0.698, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022
Epoch: 501, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.710, Val AUC_ROC Weighted : 0.710, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022
Epoch: 601, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.711, Val AUC_ROC Weighted : 0.711, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022
Epoch: 701, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.712, Val AUC_ROC Weighted : 0.712, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022
Epoch: 801, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC Macro: 0.712, Val AUC_ROC Weighted : 0.712, Val Recall Macro: 0.500, Val Precision Weighted : 0.957, Val Ham Loss : 0.022

Epoch: 901, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC
Macro: 0.711, Val AUC_ROC Weighted : 0.711, Val Recall Macro: 0.500, Val
Precision Weighted : 0.957, Val Ham Loss : 0.022
Epoch: 1001, Train loss: 0.002, Val loss: 0.002, Val Acc: 0.978, Val AUC_ROC
Macro: 0.711, Val AUC_ROC Weighted : 0.711, Val Recall Macro: 0.500, Val
Precision Weighted : 0.957, Val Ham Loss : 0.022

[]: